Proper Noun Semantic Clustering using Bag-Of-Vectors

Ali-Reza Ebadat, Vincent Claveau, Pascale Sébillot

Proper Noun Semantic Clustering using Bag-of-Vectors

Ali Reza Ebadat
INRIA-INSA
ali_reza.ebadat
@inria.fr

Vincent Claveau
IRISA-CNRS
vincent.claveau
@irisa.fr

Pascale Sébillot
IRISA-INSA
pascale.sebillot
@irisa.fr

Abstract

In this paper, we propose a model for semantic clustering of entities extracted from a text, and we apply it to a Proper Noun classification task. This model is based on a new method to compute the similarity between the entities. Indeed, the classical way of calculating similarity is to build a feature vector or Bag-of-Features for each entity and then use classical similarity functions like cosine. In practice, the features are contextual ones, such as words around the different occurrences of each entity.

Here, we propose to use an alternative representation for entities, called Bag-of-Vectors, or Bag-of-Bags-of-Features. In this new model, each entity is not defined as a unique vector but as a set of vectors, in which each vector is built based on the contextual features of one occurrence of the entity. In order to use Bag-of-Vectors for clustering, we introduce new versions of classical similarity functions such as Cosine and Scalar Products.

Experimentally, we show that the Bag-of-Vectors representation always improve the clustering results compared to classical Bag-of-Features representations. [1]

1 Introduction

Clustering entities extracted from texts, such as proper nouns, is a task very close to Named Entity Recognition (NER). Indeed, the goal in Named Entity Recognition is to locate and classify Named Entities (NE) into predefined groups such as Person, Location and Organization names. Locating and classifying could be done either in one step or in two consecutive steps, but most NER systems rely on supervised models, trained on manually tagged data. Yet, in this work, our goal is slightly different from this strict definition since we aim at building classes of entities without any supervision or presupposition about the classes. More precisely, we want to group proper nouns (PN) into different clusters based on their similarities. A good clustering should produce have higher similarities among PN within the cluster and less similarities between clusters.

As for any clustering problem, describing (representing the entities) and comparing (computing similarities between the representations) are crucial elements. A good clustering model is expected to show high similarities among the entities within a cluster and low similarities between entities from different clusters. The choice of the similarity function is highly dependent on the representation used to describe the entities. In this paper, we investigate the use of a new representation which is expected to outperform the standard representation commonly used. Indeed, the classical way of calculating similarity is to build a feature vector or Bag-of-Features for each entity and then use classical similarity functions like cosine. In practice, the features are contextual ones, such as words or ngrams around the different occurrences of each entity. Here, we propose to use an alternative representation for entities, called Bag-of-Vectors, or Bag-of-Bags-of-Features. In this new model, each entity is not defined as a unique vector but as a set of vectors, in which each vector is built based on the contextual features (surrounding words or ngrams) of one occurrence of the entity. The usual similarity or distance functions including Cosine and Euclidean distances, can be easily extended to handle this new representation. These various representation schemes and distances are evaluated on a proper noun clustering task.

In the next section, we review related work in the Named Entity Recognition domain. The different representation schemes, including the Bag-of-Vectors one, are detailed in Section 3, and their use to compute similarities and finally cluster the entities is presented in Section 4. Experiments are then reported in Section 5 for different similarity functions and feature vectors models. Finally, the conclusions are described in section 6.

2 Related Work

Extracting and categorizing entities from texts has been widely studied in the framework of Named Entity Recognition. The history of NER goes back to twenty years ago; at that time, its goal was to "extract and recognize [company] names" (Nadeau and Sekine, 2007). NER is now commonly seen as the task of labeling (classifying) proper noun or expressions into broad subgroups, such as person, location, organization names, etc. (Sang, Erik, and De Meulder, 2003), or more recently into fine grain groups (eg. a location can be a city, a state or a country...) (Fleischman and Hovy, 2002; Ekbal et al., 2010).

[1]This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

Several models are used for NER which could be considered in three main groups. Supervised models which need annotated data to train a supervised machine learning algorithm such as Support Vector Machine (Isozaki and Kazawa, 2002; Takeuchi and Collier, 2002), Conditional Random Field (McCallum and Li, 2003; Sobhana N.V, 2010), Maximum Entropy (Chieu and Ng, 2002) and Hidden Markov Model (Zhou and Su, 2002). In these NER models, the quality of the final results chiefly depends on the size of the training data. Semi-supervised machine learning has also been explored when the annotated data is small or non existent. Different models have been studied under this category including rule-based system (Liao and Veeramachaneni, 2009) in which simple rules help to build some annotated data, then a CRF classifier ,trained on the training data, generates new training data for the next learning iteration. Kozareva (2006) used some clue words in order to build the gazetteer lists from unlabeled data; this lit is then used to train different NER systems.

Whether supervised or semi-supervised, these approaches relies on predefined group of entities (and the corresponding training data). Yet, in a context of information discovery, defining the interesting NE categories requires deep knowledge of the domain and biases the systems since they focus on these categories and may miss interesting information. To the best of our knowledge, there is not pure unsupervised NER system. Some systems claim to be unsupervised but either rely on hand-coded rules (Collins and Singer, 1999), or external resources such as Wikipedia (Kazama and Torisawa, 2007).

From a technical point of view, similarity of complex objects (graphs, trees...) has been widely explored. The Bag-of-Vectors representation that we propose to investigate in this paper is inspired from the bag-of-bags used for image classification with SVM (Gosselin, Cord, and Philipp-Foliguet, 2007).

3 Representing entities with Bag-of-Features and Bag-of-Vectors

In our clustering task, we focus on proper nouns (PN) contained in French football reports. The texts are Pars-of-Speech tagged using TreeTagger (Schmid, 1995), and the PN are simply collected based on their tagges. In order to cluster them, we need to represent these PN so that similarities can be computed between them. As it was previously explained, vectorial representation is commonly used for this type of task: a PN is represented by one contextual vector. In this paper we investigate the use of a new representation scheme, the Bag-of-Vector, in which a PN is represented by several contextual vectors. In the remaining of this section, we first explain which contextual features, common to these two representation, are used, and then successively present the Bag-of-Features and Bag-of-Vectors approaches.

3.1 Contextual Features

Different contextual features were explored for our experiments, based on words, lemmas or ngrams surrounding each occurrences of a PN. In the experiments reported in this paper, we only present the results for the features that yielded the best results. These are based on 3-grams collected in a window of 4 tokens before and after each PN occurrence in the sentence which are linearly combined with lower ngram (n=2,1) in order to cover data sparsity. An example of collected n-grams is given in Table 1.

Sentence	
Zigic donne quelques frayeurs à Gallas et consorts en contrôlant un ballon chaud à gauche des 16 mètres au devant du Gunner.	
PN	**ngram feature**
Zigic	donne quelques frayeurs — quelques frayeurs à
Gallas	donne quelques frayeurs — quelques frayeurs à, et consorts en — consorts en contrôlant
Gunner	mètres au devant — au devant du

Table 1: ngram features for proper noun N=3, W=4

Different weighting schemes for the collected ngrams were also explored, in order to give less importance to very common ngrams. Here again, we only present the one giving the best results, which is a standard TF-IDF (note that in a short window, TF is almost always equal to 1, the weighting scheme is thus mostly a pure IDF).

For those PN which don't have any common 3-gram with other PN, lower ngrams are useful to make some (weak) connections with other PN. Finally, a linear combination of IDF for different n is defined as final weight score for a given PN.

3.2 Bag-of-Features (BoF)

In the standard BoF model, for each detected PN in the corpus, a single (weighted) feature vector is simply built based on the ngrams before and after the PN occurrences in the whole corpus. Thanks to its sparsity, the resulting vector allows very effective distance computation. Yet, in such a representation, the ngrams coming from the different occurrences of a PN are mixed (added). Thus, based on this representation, the comparison of two PN cannot be made at the occurrence level. The Bag-of-Vectors representation that we propose to use, is aimed at keeping the good properties of the vectorial representation, while offering a occurrence-based representation.

3.3 Bag-of-Vectors (BoV)

In this model, each PN in the text is represented with a bag of vectors in which each vector is a standard BoF for each occurrence of the PN (see figure 1). Let consider a PN as P_1, we define a BoV as explained in equation 1.

$$BoV(P_1) = \{b_{11}, b_{12} \dots b_{1i} \dots b_{1r}\} \quad (1)$$

where P_1 is the BoV of a PN in the corpus and r is number of occurrence of P_1 as a PN in the corpus and b_{1i} is BoF of P_1 in a sentence.

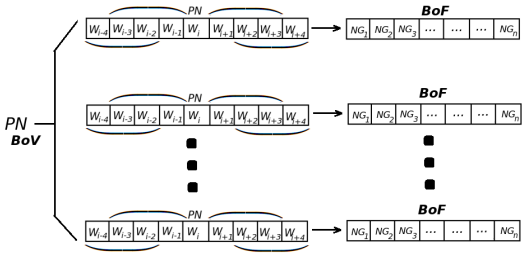

Figure 1: Bag-of-Vectors ngram for PN

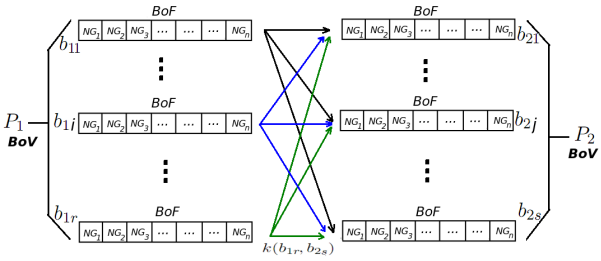

Figure 2: Similarity function on BoV

4 Similarity Functions and Clustering

This section is divided into two parts. First, we detail the similarity functions designed to handle the representation schemes presented in the previous section. Secondly, we present the clustering algorithm making the most of these similarities to build the PN clusters.

4.1 Similarity Functions

Many different similarity (or distance) functions can be used with a usual vectorial representation (that is, in our case the BoF representation). In this paper, we use three classic similarity functions: Cosine Scalar Product. In addition to these classic similarity functions, we also propose Power Scalar Product as detailed in equation 2 and call it Power Scalar. With X and Y two vectors (BoF), it is defined as:

$$\text{Power-Scalar}(X,Y) = \left(\sum_{i=1}^{n}(x_i \cdot y_i)^p\right)^{1/p} \quad (2)$$
$$X = (x_1, x_2, ..., x_n), Y = (y_1, y_2, ..., y_n)$$

The intuition behind this new similarity function is to have a discriminative scalar product by increasing the parameter p. Clearly, equation 2 is the same as Scalar Product when $p = 1$.

Those classical similarity functions work with BoF. In order to use those similarity functions with BoV, one needs to generalize them. The simplest strategy is to define a way to aggregate similarities computed from the multiple vectors in the BoV using usual similarity functions. For instance, based on the work of Gosselin, Cord, and Philipp-Foliguet (2007), one can define the similarity between two PN based on their BoV as the sum of similarity among all BoF for both PN (see figure 2). Of course, many different ways can be used to define the general similarity function such as sum-of-max or sum-of-sum of similarity. In this paper, we use both sum-of-sum and sum-of-max definitions which are formulated in Eq. 3 and 4 where $P_1 = \{b_{11}, b_{12} \dots b_{1i} \dots b_{1r}\}$ and b_{1i} is a BoF of P_1 and $P_2 = \{b_{21}, b_{22} \dots b_{2j} \dots b_{2s}\}$ and b_{2j} is a BoF of P_2. In Eq. 3, k could be any similarity function and r , s are the number of BoF contained in P_1's and P_2's BoV standard.

$$Sim_{SS}(P_1, P_2) = \sum_{i=1}^{r}\sum_{j=1}^{s} k(b_{1i}, b_{2j}) \quad (3)$$

$$Sim_{SM}(P_1, P_2) = \sum_{i=1}^{r} \max_{j} k(b_{1i}, b_{2j}) \quad (4)$$

In equation 3 and 4, the complexity depends on r and s as number of instances of the first and the second PN. In addition, the complexity of $k(b_{1i}, b_{2j})$ has to be considered. For both equations computational cost is $O(r * s * n)$, where n is length of feature vector. But this complexity remains very low since each BoF is very sparse (even sparser than the unique BoF that is used in the standard representation). Indeed, for sparse data the computational cost of $k(b_{1i}, b_{2j})$ only depends on non-zero components of the vector for Cosine, Jaccard and Power Scalar similarity functions.

Power kernel

Extending this idea in a Support Vector Machine context, Gosselin, Cord, and Philipp-Foliguet (2007) also proposed the so-called Power Kernel in order to increase the higher values and decrease lower values. As it can of course be considered as a similarity function, we also experiment with a generalized similarity function with Power Kernel defined in equation 5, in order to build a discriminative similarity function. In this equation, when $q = 1$ the equation is the same as equation 3 for generalized similarity function.

$$Sim_{SSPK}(P_1, P_2) = \left(\sum_{i=1}^{r}\sum_{j=1}^{s} k(b_{1i}, b_{2j})^q\right)^{1/q} \quad (5)$$

$$Sim_{SMPK}(P_1, P_2) = \left(\sum_{i=1}^{r} \max_{j=1}^{s} k(b_{1i}, b_{2j})^q\right)^{1/q} \quad (6)$$

4.2 Markov Clustering

Generally, clustering is the task of assigning a set of objects into groups called clusters so that the objects within the same cluster are more similar to each other than to the objects in any other clusters. In our case, our PN clustering task can be seen as a graph clustering in which each node in the graph is a PN and an edge is a relation between two PN. In practice, this relation is defined as the similarity between PN, based on the common contextual features of their occurrences.

Among all the possible clustering algorithm, we thus decided to use Markov Clustering Algorithm (MCL) which

was first proposed as a graph clustering algorithm (van Dongen, 2000). It also offers an interesting advantage over more classic algorithms like k-means or k-medoids in that MCL does not require the user to specify the expected number of clusters.

MCL is a clustering algorithm which simulates Random Walk within a graph represented as the similarity matrix. It only relies on two simple operations - expansion and inflation. Each entry in row_i and col_j, is the similarity between PN_i and PN_j. *Expansion operation* is a simple matrix multiplication operation which makes a new connection between nodes without direct edge and make other edges stronger. Expansion helps the algorithm to make the similarity with in the (potential) cluster stronger; *Inflation operation* is defined as the similarity matrix entry, power to a inflation rate with a normalization of the columns in the matrix. Inflation helps the algorithm to separate clusters from each other. In this paper, we use a fixed inflation rate (1.5) as proposed by MCL developers.

In MCL, these two operations are applied consecutively until there is no more change in the matrix. The final matrix is then used to find the clusters: each cluster is a group of columns in the final matrix which have almost the same values. For our experiments, we used a Perl implementation of MCL called minimcl obtained form http://micans.org/mcl.

5 Experiments

The previously defined representations and similarity functions with Markov Clustering Algorithm (MCL) are used to cluster PN in football reports. In this section, we first explain the evaluation metrics used, the experimental data, and then the results with different similarity functions are explained.

5.1 Evaluation Metrics

The goal of the clustering is to have high intra-cluster similarity (similar objects in same cluster) and low inter-cluster similarity (objects from different cluster are dissimilar) which is called internal criterion. But having a good score on an internal criterion doesn't mean necessarily a good effectiveness. One way is to use a ground truth to find out how much the clustering results are similar to it, which is called external criterion (Manning, Raghavan, and Schütze, 2008).

Different metrics of cluster evaluation (or comparison) such as Purity or Random Index (Rand, 1971) have been proposed in the literature. Yet, these metrics are known to be not very discriminative, sometimes being over-optimistic, especially when the number of members in each cluster is relatively small (Vinh, Epps, and Bailey, 2010). To the contrary, Adjusted Random Index (ARI) (Hubert and Arabie, 1985) is known to be robust as it is an adjusted-for-chance form of the Rand index. It is chosen as the main evaluation metric in this paper.

5.2 Data

In this experiment, we use specific football reports called minute-by-minute report which were extracted from French specialized websites. Almost each minute of the football match is summarized for the important events during that minute, including player replacement, fouls or goals (see table 2).

Minute	Report
80	Zigic donne quelques frayeurs à Gallas et consorts en contrôlant un ballon chaud à gauche des 16 mètres au devant du Gunner. Le Valencian se trompe dans son contrôle et la France peut souffler.
82	Changement opéré par Raymond Domenech avec l'entrée d'Alou Diarra à la place de Sidney Govou,pour les dernières minutes. Une manière decolmater les brêches actuelles?

Table 2: Minute-by-minute football report in French

Cluster label	N	Of total
player	712	68%
team	114	11%
town	62	6%
trainer	44	4%
other	43	4%
country	26	2%
championship	26	2%
stadium	13	1%
referee	11	1%

Table 3: NE classes in ground truth

For the experiments reported below, 4 football matches were considered; it corresponds to 819 sentences, 12155 words and 1163 occurrences of PN (198 unique PN). In order to build a ground truth, one person specialized in football match annotation was asked to manually cluster the PN of these match reports. It resulted in 9 ground-truth clusters for PN, including player name, coach name, etc., which are listed in Table 3. Unsurprisingly, the most frequent PN in the report are player name, which could make this class important to our model. It is also interesting to see how unbalance these ground-truth cluster are.

5.3 Results

In this experiment, we evaluate three different models on PN clustering; Bag-of-Features, Bag-of-Vectors and combination of BoV with Power Kernel. For all models, we use the Cosine, Scalar Product and Power Scalar similarity functions. With all three models, we utilize Markov Clustering Algorithm (Inflation Rate=1.5). We run the model with sum-of-sum and sum-of-max similarity functions on BoV features. For all similarity functions, we also report the results for classic BoF. In addition to this, we also perform a random clustering of the PN as a baseline. All the results are presented in Table 4.

For all of results in Table 4, there are 198 PN in 8 or 9 clusters in the final results. One of the main result which is worth noting is that BoV improved the Cosine results, while

Similarity	**BoF**	BoV_{SS}	BoV_{SSPK}
Cosine	6.91	25.81	-
Scalar Product	38.27	39.24	37.32
Power Scalar	40.08	39.24	42.76

Table 4: Similarity functions comparison with sum-of-sum, in terms of ARI (%)

Similarity	**BoF**	BoV_{SM}	BoV_{SMPK}
Cosine	6.91	29.56	-
Scalar Product	38.27	22.05	25.80
Power Scalar	40.08	35.35	37.18

Table 5: Similarity functions comparison for sum-of-max on BoV, in ARI (%)

in all cases BoV with Power Kernel (BoV_{SSPK}) outperformed standard BoF and BoV representation. Cosine similarity with Power Kernel could not cluster all PN in which in final results there were only 50% of all PN. The maximum ARI is obtained with Power Scalar ($p = 2$) when combined with Power Kernel ($q = 2$, other q gives slightly inferior but comparable results).

In addition to the sum-of-sum generalized similarity function, we also examine sum-of-max (see Eq. 4). The results are listed in Table 5 and show that sum-of-sum similarity made slightly better clusters with different similarity functions except for Cosine. But, these results are still far better than the usual BoF ones.

Sum-of-max similarity function didn't show improvement for Cosine, scalar and Power Scalar similarity function. Comparing results in Table 4 and Table 5 shows that the number of connected (similar) PN is an important factor in final results. In sum-of-sum, all connections are considered in final similarity calculation while in sum-of-max, only connections with maximum similarity for each PN are used.

5.4 Error Analysis

BoV with ngram feature seems a good model for clustering entities, obtaining very high results, but it is interesting to have a closer look at the causes of errors in the final clustering results. To do so, we examine the errors for each class in the ground truth, and we are also interested to know what are the PN that cannot be clustered with our model and why.

First we calculate the precision and recall for each PN in the clusters (Bagga and Baldwin, 1998). which is formulated in equation 7, in which PN_i is i^{th} NE in cluster C_j and $L(PN_i)$ denotes the class of PN_i.

$$Pre(NE_i, C_j) = \frac{|L(NE_i) \cap C_j|}{|C_j|} \quad (7)$$

Then we compute the average precision for each class in the ground-truth, which is the average precision of its members.

For our best model (a combination of Power Scalar with Power Kernel), the precision, recall and F-measure are reported in Table 6.

class	Precision	Recall	F-Measure
player	74.87	76.52	75.68
referee	52.91	53.12	53.02
trainer	24.30	23.61	23.95
town	21.50	19.00	20.17
team	15.36	30.56	20.44
other	9.74	24.00	13.85
country	9.26	37.50	14.85
championship	4.53	50.00	8.31
stadium	3.93	62.50	7.39

Table 6: Class average precision for best model

The best f-measure is for the player name class which is also the most important class in the report (because of the player names frequency in the report, see Table 3).

The class evaluation also shows that "stadium" is the most difficult class to cluster in this model. We found that ngrams around "stadium" NE in the report are spread out in the report and near to other PN which makes the clustering difficult for this class because of low similarity between them.

It is also interesting to note that we use a simple PN detection technique solely based on the Part-of-Speech and it causes some errors. For example, "Guingampais" is guessed as a Proper Noun by TreeTagger (which does not have this word in its lexicon) which is not true. Moreover, it also bias the ngrams counts and thus the IDF used for the description of the other PN. Conversely, no PN from the ground-truth is missing from the automatic clustering results. This simple detecting system has thus a sufficiently good recall and decent precision for this application.

6 Conclusion and Future Work

In this paper, we tackled an unsupervised text mining problem: we proposed a model for entity clustering based on the use of new representation schemes called Bag-of-vectors. This representation keeps the effectiveness of the vectorial representation, and thus allows an fast and easy calculation of distances, while representing each occurrence of entity independently. In order to compute these distances, we have shown that simple generalizations of the usual vectorial similarity functions can be made. The whole approach, evaluated on a proper nouns clustering task in the football domain, outperformed the standard approach. In particular, the new Power-scalar similarity function that we proposed, combined with the Power-Kernel generalization allowed us to build a very discriminative model.

There are some other aspects of this problem that we are interested to tackle in the future. First of all, From an applicative point of view, we are also interested to cluster NE in transcribed text of football reports. In the transcribed text, there are different kinds of noise such as misspelled NE or some non word tokens. We are interested to see how robust our model is against noisy data. Another applicative foreseen work is to use this type of BoV representation in information retrieval in which documents are often represented as Bag-of-Words.

From a more fundamental point of view, many other similarity functions and many other ways to generalize them for BoV can be proposed. For instance, here we only used the maximum and the sum to aggregate the different vector similarities, and both can be seen as OR logical operator. Fuzzy logic offers many other logic operators to model the OR (T-conorms), and more generally many aggregation operators with well controlled properties that could be intersting to test in this context.

References

Bagga, A., and Baldwin, B. 1998. Entity-based cross-document coreferencing using the vector space model. In *proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, 79–85. Stroudsburg, PA, USA: Association for Computational Linguistics.

Chieu, H. L., and Ng, H. T. 2002. Named entity recognition: a maximum entropy approach using global information. In *proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, 1–7. Stroudsburg, PA, USA: Association for Computational Linguistics.

Collins, M., and Singer, Y. 1999. Unsupervised models for named entity classification. In *proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Ekbal, A.; Sourjikova, E.; Frank, A.; and Ponzetto, S. P. 2010. Assessing the challenge of fine-grained named entity recognition and classification. In *proceedings of the 2010 Named Entities Workshop*, NEWS 10, 93–101. Stroudsburg, PA, USA: Association for Computational Linguistics.

Fleischman, M., and Hovy, E. 2002. Fine grained classification of named entities. In *proceedings of the 19th International Conference on Computational Linguistics*, 1–7.

Gosselin, P.; Cord, M.; and Philipp-Foliguet, S. 2007. Kernels on bags of fuzzy regions for fast object retrieval. In *image processing, 2007. ICIP 2007. IEEE International Conference on*, volume 1, 177–180.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classiffication*.

Isozaki, H., and Kazawa, H. 2002. Efficient support vector classifiers for named entity recognition. In *proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, 1–7. Stroudsburg, PA, USA: Association for Computational Linguistics.

Kazama, J., and Torisawa, K. 2007. Eexploiting wikipedia as external knowledge for named entity recognition. In *proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 698–707. Prague: Association for Computational Linguistics.

Kozareva, Z. 2006. Bootstrapping named entity recognition with automatically generated gazetteer lists. In *proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 15–21.

Liao, W., and Veeramachaneni, S. 2009. A simple semi-supervised algorithm for named entity recognition. In *proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*, 58–65. Boulder, Colorad: Association for Computational Linguistics.

Manning, C.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press.

McCallum, A., and Li, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, 188–191. Stroudsburg, PA, USA: Association for Computational Linguistics.

Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30:3–26.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. 66(336):pp. 846–850.

Sang, T. K.; Erik, F.; and De Meulder, F. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, 142–147. Stroudsburg, PA, USA: Association for Computational Linguistics.

Schmid, H. 1995. Probabilistic part-of-speech tagging using decision trees. In *international Conference on New Methods in Language Processing*, 44–49.

Sobhana N.V, Pabitra Mitra, S. G. 2010. Conditional random field based named entity recognition in geological text. *International Journal of Computer Applications*.

Takeuchi, K., and Collier, N. 2002. Use of support vector machines in extended named entity recognition. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, 1–7. Stroudsburg, PA, USA: Association for Computational Linguistics.

van Dongen, S. 2000. *Graph Clustering by Flow Simulation*. Ph.D. Dissertation, University of Utrecht.

Vinh, N. X.; Epps, J.; and Bailey, J. 2010. Information theoretic measures for clusterings comparison. *Journal of Machine Learning Research*.

Zhou, G., and Su, J. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, 473–480. Stroudsburg, PA, USA: Association for Computational Linguistics.